Decsion Tree

In [1]: import seaborn as sns
 from sklearn.preprocessing import LabelEncoder
 from pandas.tools.plotting import parallel_coordinates, andrews_curves
 import matplotlib.pyplot as plt
 import pydotplus
 import urllib.request

%matplotlib inline

In [3]: import numpy as np
 import pandas as pd
 names = ['Company','Open','High','Low','Close','Volume']
 data = pd.read_csv('stockpredictions.csv', names=names)
 print(data.shape)

(513, 6)

In [4]: data.sample()

Out[4]:

	Company	Open	High	Low	Close	Volume
113	ABC	23.75	23.855	23.53	23.81	23858

In [5]: data.head()

Out[5]:

	Company	Open	High	Low	Close	Volume
0	AA	11.32	11.34	10.87	11.00	585630
1	AA	11.01	11.11	10.82	10.91	258446
2	AA	10.95	10.95	10.61	10.84	233940
3	AA	10.72	10.75	10.41	10.41	282508
4	AA	10.50	10.68	10.31	10.58	267170

In [6]: data.tail()

Out[6]:

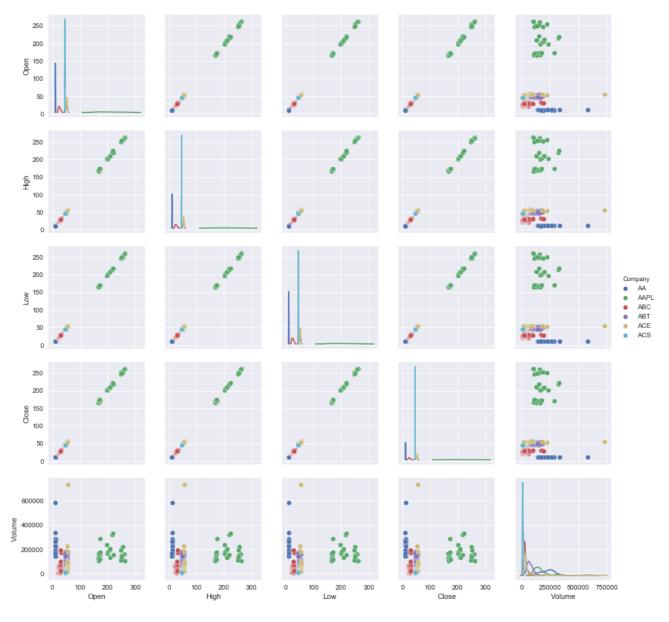
	Company	Open	High	Low	Close	Volume
508	ACE	52.91	53.545	52.91	53.51	17228
509	ACS	45.08	45.330	44.40	45.32	17241
510	ACS	45.55	45.950	45.40	45.60	9765
511	ACS	45.87	45.930	45.45	45.49	8212
512	ACS	45.53	45.880	45.36	45.49	11869

In [9]: sns.pairplot(data, hue = 'Company', diag_kind="kde")

C:\Users\Jatan\Anaconda3\lib\site-packages\statsmodels\nonparametric\kdetools.py:20:
VisibleDeprecationWarning: using a non-integer number instead of an integer will res
ult in an error in the future

 $y = X[:m/2+1] + np.r_[0,X[m/2+1:],0]*1j$

Out[9]: <seaborn.axisgrid.PairGrid at 0x8383e05a20>



In [10]: encoder = LabelEncoder()
data['Company'] = encoder.fit_transform(data['Company'])

In [11]: data.sample()

Out[11]:

	Company	Open	High	Low	Close	Volume
199	2	29.82	30.085	29.65	29.97	28542

In [12]: data.head()

Out[12]:

		Company	Open	High	Low	Close	Volume
	0	0	11.32	11.34	10.87	11.00	585630
	1	0	11.01	11.11	10.82	10.91	258446
	2	0	10.95	10.95	10.61	10.84	233940
	3	0	10.72	10.75	10.41	10.41	282508
	4	0	10.50	10.68	10.31	10.58	267170

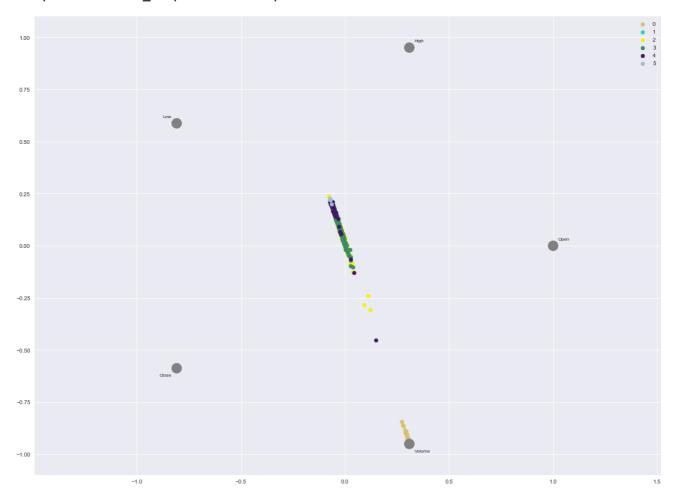
In [13]: data.tail()

Out[13]:

	Company	Open	High	Low	Close	Volume
508	4	52.91	53.545	52.91	53.51	17228
509	5	45.08	45.330	44.40	45.32	17241
510	5	45.55	45.950	45.40	45.60	9765
511	5	45.87	45.930	45.45	45.49	8212
512	5	45.53	45.880	45.36	45.49	11869

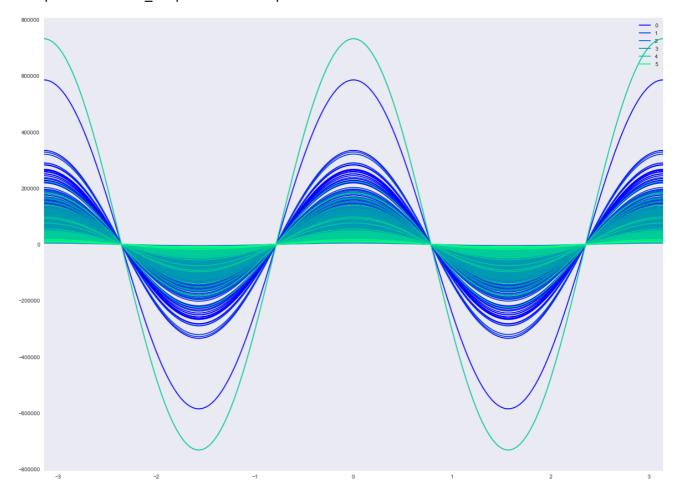
In [14]: width, height = 20, 15
 size = width, height
 plt.figure(figsize = size)
 from pandas.tools.plotting import radviz
 radviz(data, 'Company')

Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x8393b2b9e8>



In [15]: plt.figure(figsize = size)
 andrews_curves(data, 'Company', colormap = 'winter')

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x8394f537f0>



In [17]: df = (data - data.min()) / (data.max() - data.min())
 df.sample(5)

Out[17]:

	Company	Open	High	Low	Close	Volume
52	0.2	0.827393	0.848846	0.829524	0.845906	0.445127
271	0.6	0.177028	0.175304	0.175472	0.175340	0.133780
335	0.6	0.174200	0.173404	0.175072	0.174704	0.094223
443	0.6	0.160859	0.159785	0.160326	0.161017	0.055207
103	0.4	0.052730	0.051665	0.051950	0.051802	0.022765

```
In [18]: from sklearn.model_selection import train_test_split
```

In [19]: train, test = train_test_split(data, train_size = 0.60)

In [20]: len(train), len(test)

Out[20]: (307, 206)

In [21]: from sklearn.tree import DecisionTreeClassifier

In [22]: features = data.columns[1:]
 target = data.columns[0]

```
model = DecisionTreeClassifier()
In [23]:
                      model = model.fit(train[features], train[target])
                      from sklearn.metrics import accuracy_score
In [24]:
In [25]:
                      predicted = model.predict(test[features])
In [26]:
                      print('Accuracy Score:', accuracy_score(test[target], predicted) * 100)
                      Accuracy Score: 93.2038834951
In [27]:
                      from sklearn.tree import export_graphviz
In [29]:
                      Class_names = list(set(encoder.inverse_transform(data[target])))
                      export_graphviz(model,out_file = 'SP.dot',
                                                                           feature_names = features,
                                                                           class_names = Class_names,
                                                                           filled = True,
                                                                           rounded = True,
                                                                           special_characters = True)
                      graph = pydotplus.graphviz.graph_from_dot_file("SP.dot")
In [30]:
                     png = graph.create_png()
                      from IPython.display import Image
In [31]:
                      Image(png)
                                                                    Close ≤ 38.955
gini = 0.6893
samples = 307
value = [20, 21, 88, 139, 37, 2]
Out[31]:
                                                                              class = ACS
                                                                       True
                                                                                       Volume ≤ 42974.0
gini = 0.4663
samples = 199
value = [0, 21, 0, 139, 37, 2]
class = ACS
                                                           Close ≤ 16.305
gini = 0.3018
                                                            samples = 108
= [20, 0, 88, 0, 0, 0]
| class = ABC
                                                                                              Low ≤ 53.775
gini = 0.2681
samples = 33
= [0, 0, 0, 3, 28, 2]
class = ABT
                                                                                                                             Open ≤ 111.715
gini = 0.3098
samples = 166
e = [0, 21, 0, 136, 9, 0]
class = ACS
                                                              Open ≤ 47.195
gini = 0.1244
samples = 30
e = [0, 0, 0, 0, 28, 2]
class = ABT
                                                                                                                             /olume ≤ 172860.0
gini = 0.1164
samples = 145
te = [0, 0, 0, 136, 9, 0]
class = ACS
                                                                                                                                                                 gini = 0.0
                                                                                                                                                          samples = 21
value = [0, 21, 0, 0, 0, 0]
class = ACE
                                                                                                                       olume ≤ 47753.5
gini = 0.0809
                                                                        gini = 0.0
samples = 28
= [0, 0, 0, 0, 28, 0]
class = ABT
                                                                                                                       samples = 142
= [0, 0, 0, 136, 6, 0]
class = ACS
                                                                                                  Volume ≤ 45540.0
gini = 0.4938
samples = 9
value = [0, 0, 0, 5, 4, 0]
class = ACS
                                                                                                                                      Close ≤ 54.995
gini = 0.0296
samples = 133
e = [0, 0, 0, 131, 2, 0]
class = ACS
                                                                                                                                      /olume ≤ 55234.0
gini = 0.0157
samples = 126
e = [0, 0, 0, 125, 1, 0]
class = ACS
                                                                                                                                                                    Volume ≤ 89553.0
gini = 0.2449
samples = 7
llue = [0, 0, 0, 6, 1, 0]
class = ACS
                                                                        High ≤ 54.765
gini = 0.2778
                                                                        samples = 6
e = [0, 0, 0, 5, 1, 0]
class = ACS
                                                                                                      olume ≤ 54836.5
gini = 0.1107
samples = 17
e = [0, 0, 0, 16, 1, 0]
class = ACS
                                     samples = 5
value = [0, 0, 0, 5, 0, 0]
class = ACS
                                                                                                                                  samples = 109
/alue = [0, 0, 0, 109, 0, 0]
class = ACS
                                                                                                                                                                                                  samples = 1
alue = [0, 0, 0, 0, 1, 0]
class = ABT
                                                                                  samples = 16
value = [0, 0, 0, 16, 0, 0]
class = ACS
                                                                                                                  samples = 1
value = [0, 0, 0, 0, 1, 0]
class = ABT
```

```
In [32]:
              model = DecisionTreeClassifier(criterion = 'entropy')
              model = model.fit(train[features],train[target])
In [33]: predicted = model.predict(test[features])
In [34]: print('Accuracy Score : ', accuracy_score(test[target], predicted)*100)
              Accuracy Score : 94.1747572816
In [35]: | ddata = export_graphviz(model,
                                                 out_file = None,
                                                 feature_names = features,
                                                 class_names = Class_names,
                                                 filled = True,
                                                 rounded = True,
                                                 special_characters = True)
In [36]: graph = pydotplus.graph_from_dot_data(ddata)
In [37]: png = graph.create_png()
In [38]:
              Image(png)
Out[38]:
                                  High ≤ 46.875
entropy = 0.9183
samples = 3
lue = [0, 0, 0, 1, 0, 2]
class = AAPL
                                                                      Open ≤ 52.015
entropy = 0.9183
samples = 6
lue = [0, 0, 0, 2, 4,
class = ABT
                                                           Volume ≤ 45540.0
entropy = 0.9183
samples = 3
ilue = [0, 0, 0, 2, 1, 0
class = ACS
                                                                                                                         Low ≤ 53.87
                                                                                                                        entropy = 1.0
samples = 2
e = [0, 0, 0, 1, 1, 0]
class = ACS
                                                                     1, 0]
                                                         s = 2
0, 2, 0, 0]
```